Sentiment Analysis: Which Systems Work Best?

Denzel Prudent, Damani Thomas, Jon Dinh

April 2023

Abstract

The purpose of this paper is to perform an experiment of various predictive machine learning models that will perform a surface-level sentiment analysis classification on a dataset of IMDB movie reviews. The polarity of a review will be determined and will be labeled as positive or negative based on the polarity. In this paper, we will show the results of a NB-DTM model, logistic regression model, SVM model, and K-Means, Decisions Tree Classifier, Random Forest, and XGBoost model when feeding a processed feature vector of consisting of Term Frequency-Inverse Document Frequency, otherwise known as TF-IDF. Furthermore, we will show that, based on our sample data of 40000 observations, the logistic regression model outperforms the others for numerous metrics on the confusion matrix such as the F-score such as the harmonic mean. All of the used models also outperform our lower-bound baseline, the Zero Rate Classifier baseline, which achieved an F1 score of 33.47% and an accuracy of 50.12%. The Pattern Polarity library was used as another baseline for our experiment, which achieved an F1 score of 76.42% and an accuracy of 76.42% Human annotation of the reviews was also used to act as an upper bound for our models, which outperformed every model, marking its validity as an upper-bound baseline for our models. We found that with minimal tuning and feature selection, we can achieve approximately a 0.90 accuracy score in predicting correct sentiment of reviews in the IMDB dataset.

1 Introduction

The practice of sentiment analysis, otherwise known as opinion mining, has been continuously researched throughout the past decades due to its capability to deliver relevant data about the sentiment of certain topics. One such topic, movie reviews, has grown in popularity as a commonly researched topic in Sentiment Analysis. It's apparent that the movie opinions of others serve to be a crucial role in our decision-making, and in the case of online movie reviews determining the sentiment of movies can greatly influence decisions on whether to spend time and money on watching a certain film. In fact, it was concluded through a movie review study that about a third of moviegoers watch a film primarily through favorable reviews (Reinstein and Snyder, 2005). Thus, the importance of using machine learning to determine the sentiment of movie reviews has become a priority for many researchers, as an accurate and automated system can analyze a large volume of reviews. An accurate system that can classify reviews in mass can possess many real-world applications from researchers to marketers.

We will be performing one of the more common classification types, the binary classification of data, where the machine determines whether the opinion of the data falls into one of two classifications, which would be a positive or negative classification in this case. In order to classify the data, our chosen models had to use extracted vectors of the data set to determine key patterns when assigning sentiment to data. The chosen feature vector for all our models was the TF-IDF vector, a feature used to classify the importance of words in a given sentence. For this experiment, we are analyzing the performance of

many common models used in NLP research, such as the NB-DTM model, logistic regression model, SVM model, and K-Means model, specifically when trained on a dataset of approximately 40,000 observations.

In this research, the main purpose of this experiment is to test each model in reading and capturing the surface layer sentiment to accurately predict polarity.

2 Related Works

Sentiment analysis (SA) and Opinion Mining (OM) have been the subject of numerous NLP studies in the past few studies. Oftentimes, researchers are primarily interested in finding the best model and hyperparameters that yield the best results for the sentiment task. Many studies have been conducted that led to the overall improvement of practical sentiment analysis and sentiment analysis research. One such study was the Pang and Lee, 2008 sentiment analysis study. It comprises of ideas implemented in this study such as term presence and frequency, which was used as the sole feature of most of our models, and also manual annotation of data, which was conducted and used as an upper baseline for our models.

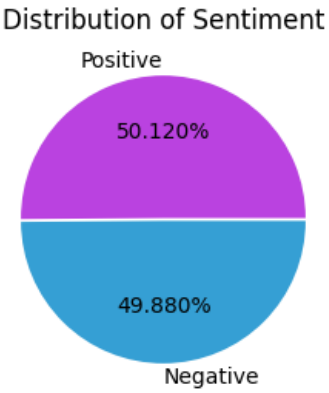
Similar to this project, there have been SA projects implemented for classifying products based on the review of consumers or experts on a specific topic. There exist SA systems designed for data on product reviews (Dave et al., 2003) and movie reviews (Pang and Lee, 2004).

Other than movies, there is also various research on other types of media, including social media sites such as Twitter, where there are many studies using datasets consisting of opinionated posts by users on these sites. One such study involves a supervised approach using lexical resources and Twitter hashtags to identify positive negative and neutral tweets (Kouloumpis et al., 2021). Although dealing with a different type of data, and classifying the data with three classes instead of two, this task faces a similar challenge to the research done in this paper - Irony. Social Media posts, Movie reviews, and many other types of user-generated content contain a large amount of irony that can skew the predictions of certain models based on how much is included in the dataset. For example, in an opinion mining study (Carvalho et al., 2009), the polarity detection rules identified negative reviews with 90% precision, but positive reviews with only 60% accuracy, with approximately 35% of the error consisting of verbal irony.

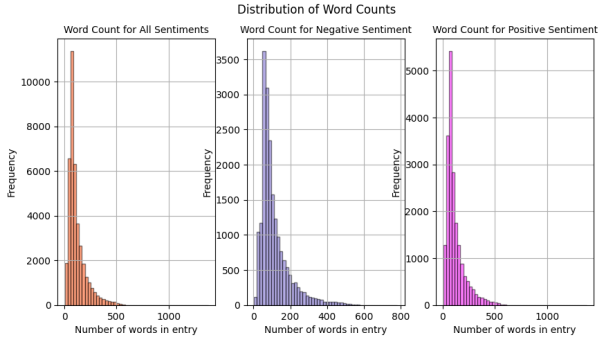
The use of many models acting as a predictor in identifying the polarity of user-generated content was performed in the Jain et al., 2017 study. The research included many models included in this paper, such as the NB, SVM, and logistic regression model (models used excessively in classification projects). Similar to this paper, the research compares the accuracy and F1 Score of these models when inputting the selected features such as TD-IDF and Count encoding.

3 Data

The necessary dataset for this project was a large set of IMDB movie reviews, with an associated polarity score. We observed many datasets used from many competitions on the machine learning platform Kaggle. Many datasets used were poorly curated, as many consisted of a low amount of data or were heavily biased in polarity. We then found the dataset used in this paper, which originally consisted of 40,000 observations of movie review data. It was curated and published by the team at Kaggle, as well as collaborated on by Machine Learning Engineer M Yasser H. There's a slight bias in the original data, as there are 20,019 negative observations and 19,981 positive ones. However, after removing duplicated entries in the data there were observed to be 39,719 entries used for research. Without these duplicates, there were 19907 positive observations and 19812 negative observations. It can be seen below that the class label distribution of positive and negative sentiment polarity is extremely even.



Furthermore, we can see from the figure below that, in addition to class labeling being balanced, the length of movie reviews is also balanced across all polarity labels.



Although scarcity is inherent in many vectorization methods and can lead to the instability of a model, the data set is large and balanced across labeled sentiment and review length. The dataset was suitable for our research as it was designed for the bi-classification of data as either positive or negative.

In order to prevent data leakage and ensure the reliable accuracy of our models, We performed an 80 : 10 : 10 split in our dataset, where 80% of the make up the training set, 10% make up the validation set, and 10% make up the testing set. It's necessary to perform a split our data in this proportion in order to have a large amount of data for our models in the training stage while also having enough testing data to perform on to ensure an accurate experiment.

4 Methodology

4.1 Dataset Pre-processing & Feature Extraction

In order to formulate a suitable input of our data for our models, we are required to pre-process the dataset, as it will allow our data to be suitable when feeding it as input in our models. As this is user-generated data, our reviews are likely to contain words and phrases that are deemed as words we shouldn't consider when creating our feature vector, otherwise known as stop-words. These are likely words that bear no significance in classifying our data as their too common and ubiquitous. In addition to removing these stopwords, we also performed these steps:

- Removed punctuation such as "?", "," and ":"
- Removed tokens less than two characters long, Ex: "a"
- Converted every character lowercase to ensure both uppercase and lowercase versions of the word was captured as a single unique token. Ex: "Terrible" and "terrible"

As the data was preprocessed and cleaned, it was suitable to formulate a feature vector from it. Feature extraction is necessary as it allows us to create suitable input data to use in our models. Our models need the input of fixed-length and numeric values in order to work, which is the reason why raw text isn't suitable for modeling input. As a solution, we convert the raw text to a feature of numbers. The models will use these extracted numeric features to formulate an estimator or predictor that will eventually predict the sentiment polarity based on the inputted vector. As mentioned earlier, the feature vector we chose for all our models in this experiment was "TF-IDF", also known as Term Frequency - Inverse Document Frequency. This feature will act as a statistic to reflect how important a word is in a review of our entire dataset. It's a very suitable feature for our sentiment classification project, as it will allow meaningful words, eg: "hate", to carry much more polarity when determining sentiment.

4.2 Logistic Regression

Logistic Regression is often used for classification and predictive tasks. It essentially estimates the probability of an event occurring. Since the model relies on probability estimates, the dependent variable is bounded within a range of 0 and 1. The estimates are then transformed using Log Odds;

$$log(odds) = Logit(P) = ln(\frac{p}{1-p})$$

The beta parameter or coefficient is commonly estimated with MLE and optimized for the best fit of Log Odds. In a binary classification task, a probability of 0.5 or less predicts a label of 0. An estimate of greater than .5 predicts a 1.

4.3 Support Vector Machines

The Support vector Machine model is robust, versatile, and highly regarded in supervised learning for classification and regression tasks due to its ability to accurately predict labels in high-dimensional spaces. It works by maximizing the margin between the positive and negative data points, then labeling the data points based on the side of the hyperplane on the data lies. Historically, it outperformed many other models in comparative sentiment analysis studies, such as O'Keefe and Koprinska, 2009

4.4 Naive Bayes

Naive Bayes bases itself upon Bayes' Theorem, which determines the probability of an event based on the existing knowledge of conditions and factors that may be related to the event within that system. The theorem goes as follows:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

where $P(A|B)$ can be determined, given that B occurs and $P(B) \neq 0$ and that $P(B|A)$ is known. For the purpose of Naive Bayes, A refers to the classes that a given review can be labeled as. B refers to the features that each review contains with these features being the determining factor in whether the review is labeled as positive or negative. However, each feature contributes independently to the probability of the label, which is why the model is referred to as "naive".

4.5 K-Nearest Neighbors

The k-means model is a supervised classification model that finds the k closest data points and finds the majority vote, or majority label, of the labels of these k data points to make a prediction on a data point inputted into the model. A larger value of k will result in a clear decision boundary but can make a false prediction of the data depending on the representation of k numbers. Conversely, a smaller k value may lead to overfitting.

4.6 DecisionTreeClassifer

Decision Tree classifiers take a rule-based approach to determine the correct class as opposed to a probabilistic approach, such as in the case with Naive Bayes. A root node that contains possible features based on the data set used to train the classifier acts as the core of the model. The root node then branches off into other nodes and splits the feature space between them based on the perceived value of the features and how they relate to the possible classes. This splitting continues until each feature is correlated with one of the classes, represented by the classification of the leaf nodes of the tree. Therefore, the depth of the tree and how many features are required for a node to spilt play a role in the accuracy of the model.

4.7 RandomForestClassifiers

Random Tree classifiers utilize the rule-based approach discussed in the section of Decision Tree classifiers, but to a further extent. To assign features to classes, Random Tree classifiers generate a number of uncorrelated decisions trees that follow the typical logic attributed to Decision Tree classifiers. The Random Tree classifier then takes the average of the class predictions for a given feature produced by the decision trees and uses the average to determine what

class those features correlate with. Using the average score of class from multiple uncorrelated trees should produce more accurate results than just using the predictions of just one tree because of the nature of independents. Errors concerning the decisions made in one tree do not affect another. Therefore, by having more trees, you reduce potential errors and biases that would result from only having one tree.

4.8 XGBoost

XGBoost classifiers, otherwise known as Extreme Gradient Boosting classifiers make use of several weaker prediction models, usually decision trees, to generate a model that has been optimized based on the models preceding it and through the use of an objective function. For each proceeding model, the residual errors that were present in the previous generation are accounted for and evaluated for the purpose of improving the accuracy of the model.

4.9 Model Baselines

4.9.1 ZeroR

For the lower bound of performance, ZeroR was utilized. Although the class distribution is relatively even, there is 0.24% difference favoring positive sentiment. Assigning all input data with the majority class provided a 0.5 accuracy and recall score, a 0.25 recall score and a 0.33 f-score.

4.9.2 Pattern Polarization

Pattern is a multi-purpose python library suited to handling NLP, data mining or machine learning tasks. The NLP function *polarity* utilizes a predetermined English dictionary of 1528 words with assigned polarity values between -1 and 1. Used on the IMDB dataset, it serves as a mid baseline model with 0.76 accuracy, precision, recall, and f-score.

4.9.3 Human Annotation

To understand how long the task of sentiment analysis would take for a human and to provide an upper bound of performance for each system tested, the development corpus was annotated. The annotation itself took roughly 17 hours to complete. It achieved an accuracy, precision, recall, and f-score of 96.90%. To get an understanding of the capacity in which the best test system performed in comparison to the annotation, the percentage of accuracy, precision, recall, and f-score between the two is calculated. The resulting percentage for each of these categories is roughly 92.76%, which raises the question of whether the increased accuracy via human annotation is worth the time given the capacity of performance.

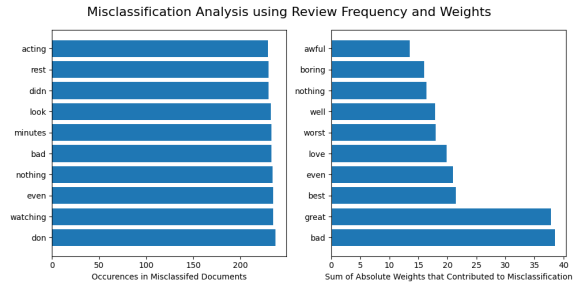
5 Experiments

First, the vectorized reviews were split by convention into training, validation and test sets with a ratio of $80 : 10 : 10$. Each set had the following sizes;

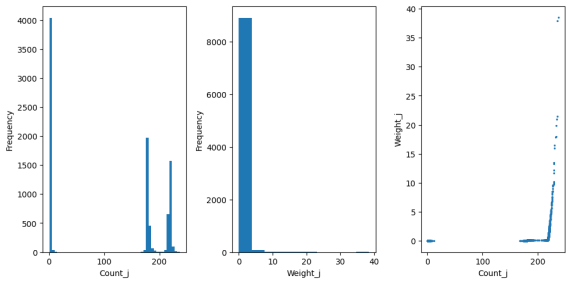
Train feature shape:	31814
Train label shape:	31814
Dev feature shape	3933
Dev label shape	3933
Test feature shape	3972
Test label shape	3972

Initial modeling was done with the baseline Naive Bayes Classifier, Logistic Regression, Support Vector Machine, K-Nearest Neighbors, Decision Tree, Random Forest and XG-Boost models with default hyperparameters and vectorization. From the results, we chose to focus on the Logistic Regression model due to it being the highest performing model and one of the least computationally expensive. [See figure]

Error analysis was conducted using Shapley Values (SHAP). Misclassified reviews classified into one of two cases. Either the actual label was 0 and the predicted label was 1, or the actual label was 1 and the predicted label was 0. The vocabulary that aided in the misclassifications were aggregated along with the number of documents they occurred in and their sum of absolute weights. An example of this data extraction can be seen in the figure;



The top 10 words that occurred in the most misclassified reviews are shown in the left plot. The right plot represents the top 10 absolute weights that contributed the wrong prediction. However, to get a holistic view of what contributes to the errors, we must view the distributions.



The first scatter plot represents the distribution of document frequencies per word. The second plot represents the distribution of sum of absolute weights per word. The final scatter plot depicts the relationship between the document frequency of a word compared to the sum weight of a word. Based on this information, it is clear that the more a word appears in these misclassified reviews, the higher the weight and therefore the more influence on the incorrect label.

Feature selection was conducted based on the rate of document frequency in incorrectly classified reviews. We found through iteration that using a threshold of 225 documents provides the highest performance. From the original vocabulary set size of $82,316$, 52 words were dropped, decreasing the total vocabulary set during vectorization to 82262.

Past error analysis, the Logistic Regression model was tuned with GridSearchCV. Max iteration was set to 200, and the scoring function was *accuracy*. We found that the best hyper-parameters were $C = 2$ and $tol = 1e - 5$, with the rest being default values.

In addition to preventing data leakage by splitting the data into train/validation/test sets, the split was conducted using stratification according to class label. Furthermore, cross-validation was conducted using 5 folds.

6 Results and Discussion

Determining the surface-level sentiment on user-generated content is a rather complex task due to the subjective nature of user-generated content. However, practical and classic Natural Language Processing methodologies such as cleaning and preprocessing data, using a classic TF-IDF feature vector to determine the importance of words, and performing the classification using a tuned Logistic Regression model can allow us to achieve this classification task with a great accuracy of 89.8% on the testing set.

The process of using feature selection to select the best features that performed well allowed the logistic regression model to increase performance on the validation set, as the accuracy of the model increased from 0.898 to 0.901 accuracy on the validation set. Based on this result, feature selection is shown to be a valid method for improving the accuracy of predictive models and should be implemented often in sentiment classification tasks. In addition, tuning the model with 5-fold cross-validation was shown to lower the performance very slightly on the validation set to 89% on the validation set. However, when combining a tuned logistic regression model with feature selection we are able to see that the model retains its high performance from the validation set to the test set as it achieved an accuracy of 89.8% on the test set. Thus, we are able to expect our model to perform consistently with new or varied data.

7 Conclusion

In this paper, we experimented with various predictive models in determining their performance in classifying user-generated movie reviews. We analyzed their performance metrics such as recall, accuracy,

and F1 score in order to identify the best-performing model. We observed that the logistic Regression model outperforms every other predictive model and the Zero-R and Pattern-Polarity library. When applied with a 5-fold cross-validation tuning and feature selection were able to create a general, valid, robust, and high-performing model that achieves an accuracy of 89.8% on the testing set.

8 Future Work

It would be worth exploring the variations of TF-IDF ad feature vectors for our predictive models. Variants such as Maximum TF normalization, sublinear TF scaling, and other document and query weighting schemes. It's possible that the variation of the weights of certain words will lead to an increased accuracy of prediction when inputted in our studied models. Similarly, it is worth experimenting with feature extraction methods other than Bag of Words.

The SHAP Python library was used as our sole method for feature selection. It's possible that a more reliable library or tool can be used for feature selection, thus it would be worthy experimenting with a further approach.

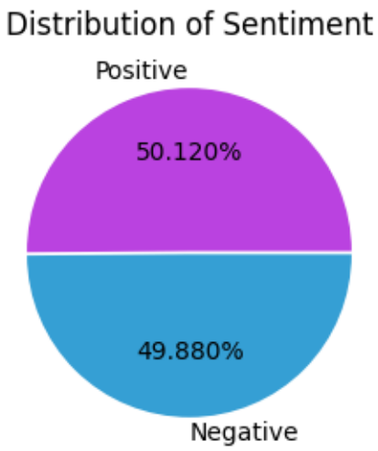
The deep-learning model known as Convolutional Long Term Short Memory (Co-LSTM) has been used in many sentiment analysis research studies where the model outperformed other predictive models or achieved a high accuracy when using a large user-generated dataset. As seen through the Behera et al., 2021 study, the Co-LSTM model achieved an accuracy of 0.9496% with an unbiased dataset of 25000 observations.

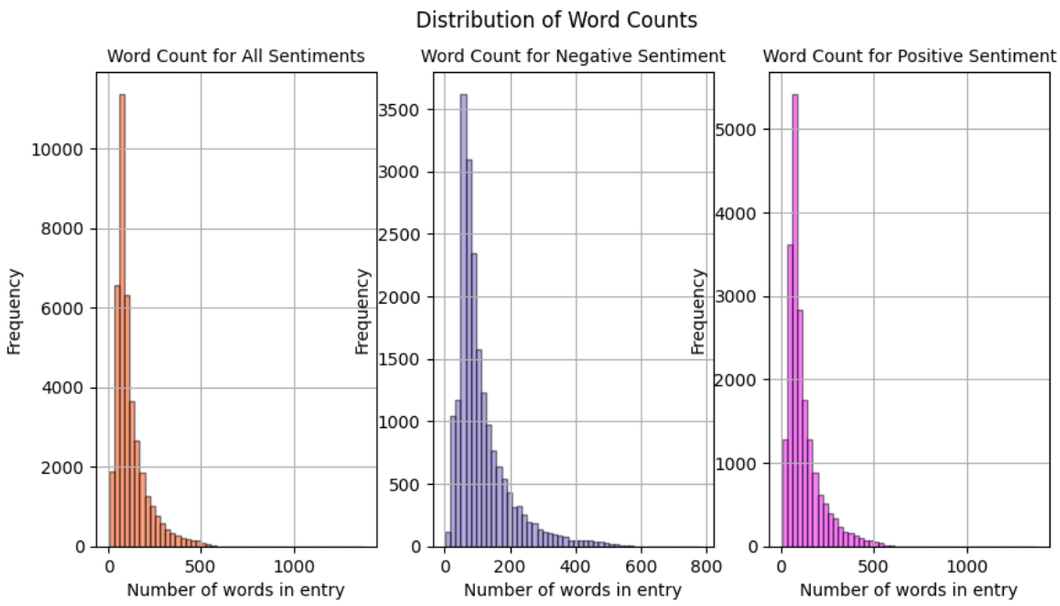
Finally, it's possible that using a pre-trained vocabulary set with predetermined weighted words would work better in our experiment that our current vectorization method. Using a pre-trained vocabulary set could decrease computation time, give faster convergence, and help eliminate ambiguous or neutral words or noise which can help produce more accurate data. Also, a published and verified vocabulary set can be more reliable and universal which can translate to a more reliable study.

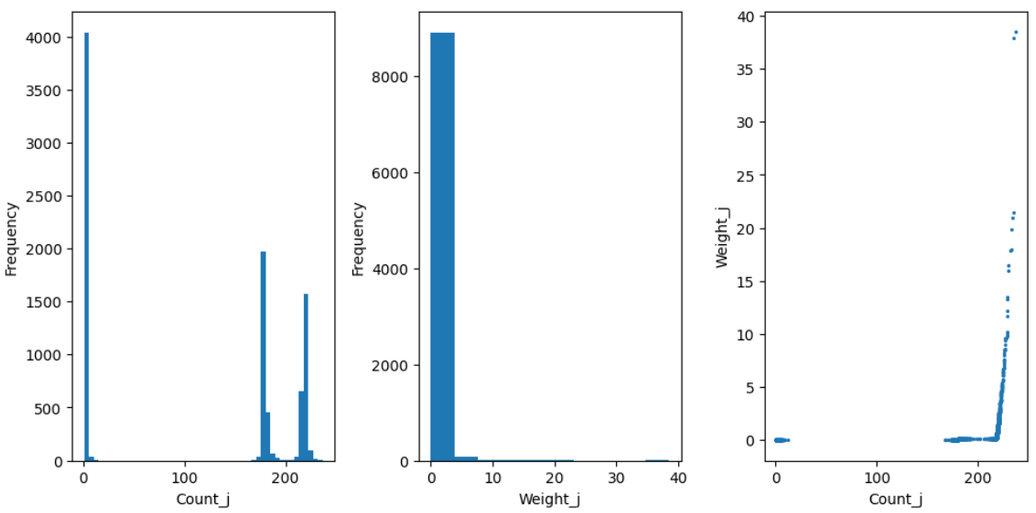
9 Contributions

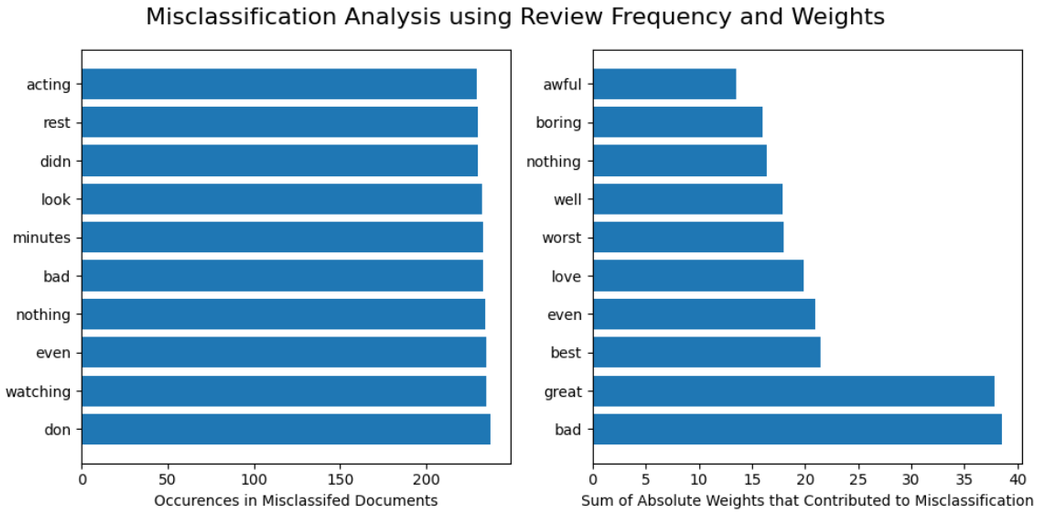
- Denzel Prudent: Author, Performed some evaluations of the models
- Damani Thomas: Author, Performed the human annotation
- Jon Dinh: Author, Engineer of Machine Learning Models

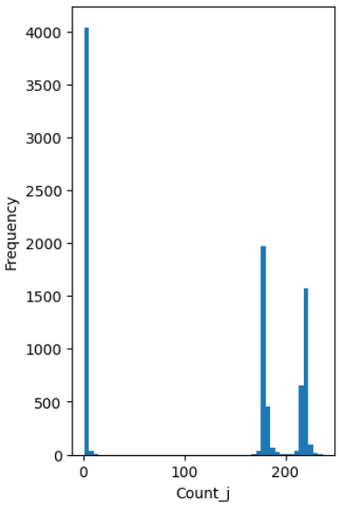
Figures

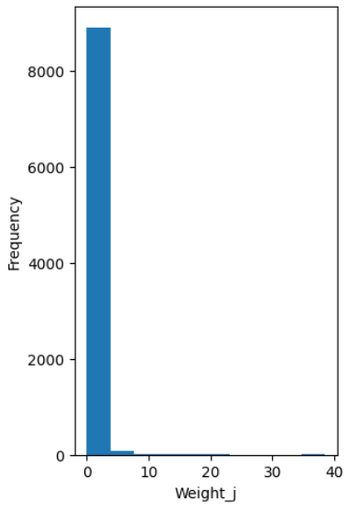

Distribution of Sentiment
Positive
50.120%
49.880%
Negative

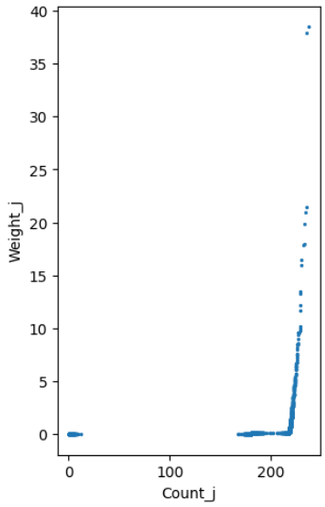

Distribution of Word Counts
Word Count for All Sentiments
Word Count for Negative Sentiment
Word Count for Positive Sentiment
Frequency
Number of words in entry


Frequency
Count_j
Weight_j


Misclassification Analysis using Review Frequency and Weights
acting
rest
didn
look
minutes
bad
nothing
even
watching
don
Occurences in Misclassifed Documents
awful
boring
nothing
well
worst
love
even
best
great
bad
Sum of Absolute Weights that Contributed to Misclassification







model	accuracy	precision	recall	f1_score	TN	FP	FN	TP	roc_acc	FPR	TPR	TNR	FNR
ZeroR (Baseline)	0.501195901	0.251197331	0.501195901	0.334662959	0	19812	0	19907	0.5	1	1	0	0
Pattern-Polarity(0.1)	0.764243813	0.76432152	0.764243813	0.764233053	15290	4522	4842	15065	0.764261734	0.228245508	0.756768976	0.771754	0.243231024
NB-TFIDF	0.872362065	0.872766341	0.872362065	0.872333014	1743	219	283	1688	0.872398633	0.111620795	0.856418062	0.888379	0.143581938
Logistic-TFIDF	0.898804983	0.898965359	0.898804983	0.898792159	1743	219	179	1792	0.89878118	0.111620795	0.909183156	0.888379	0.090816844
SVM-Linear-TFIDF	0.895499619	0.895534233	0.895499619	0.895496024	1747	215	196	1775	0.895488017	0.109582059	0.900558092	0.890418	0.099441908
KNN-TFIDF	0.778286295	0.785024137	0.778286295	0.776913765	1374	588	284	1687	0.778108258	0.29969419	0.855910705	0.700306	0.144089295
DecisionTree-TFIDF	0.725400458	0.725478719	0.725400458	0.725385368	1439	523	557	1414	0.725418802	0.26656473	0.717402334	0.733435	0.282597666
RF-TFIDF	0.858886346	0.859081748	0.858886346	0.858871566	1707	255	300	1671	0.85891179	0.129969419	0.847792998	0.870031	0.152207002
XGB-TFIDF	0.861683193	0.861927925	0.861683193	0.861654823	1664	298	246	1725	0.861652214	0.151885831	0.875190259	0.848114	0.124809741
Logistic-TFIDF-FeatSHAP	0.901093313	0.901155274	0.901093313	0.901087878	1755	207	182	1789	0.901078249	0.105504587	0.907661086	0.894495	0.092338914
Logistic-TFIDF-FeatSHAP-Tuned	0.890735146	0.890872998	0.890735146	0.890722512	1745	236	198	1793	0.890710367	0.119131752	0.900552486	0.880868	0.099447514
Logistic-Final	0.897532729	0.897713609	0.897532729	0.897517925	1756	225	182	1809	0.897504824	0.113579001	0.908588649	0.886421	0.091411351

References

Behera, R. K., Jena, M., Rath, S. K., & Misra, S. (2021). Co-lstm: Convolutional lstm model for sentiment analysis in social big data. *Information Processing Management*, *58*(1), 102435. https://doi.org/https://doi.org/10.1016/j.ipm.2020.102435

Carvalho, P. C., Sarmento, L., Silva, M. J., & de Oliveira, E. (2009). Clues for Detecting Irony in User-Generated Contents:Oh...!! It's "so easy" ;-). *Text Sentiment Analysis (TSA'09)*. https://doi.org/10.1145/1651461.1651471

Dave, K., Lawrence, S., & Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. *Proceedings of the 12th International Conference on World Wide Web*, 519–528. https://doi.org/10.1145/775152.775226

Jain, S., Malviya, S., Mishra, R., & Tiwary, U. S. (2017). Sentiment analysis: An empirical comparative study of various machine learning approaches. *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, 112–121. https://aclanthology.org/W17-7515

Kouloumpis, E., Wilson, T., & Moore, J. (2021). Twitter sentiment analysis: The good the bad and the omg! *Proceedings of the International AAAI Conference on Web and Social Media*, *5*(1), 538–541. https://doi.org/10.1609/icwsm.v5i1.14185

O'Keefe, T., & Koprinska, I. (2009). Feature selection and weighting methods in sentiment analysis.

Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, *2*(1–2), 1–135. https://doi.org/10.1561/1500000011

Reinstein, D., & Snyder, C. (2005). The influence of expert reviews on consumer demand for experience goods: A case study of movie critics. *Journal of Industrial Economics*, *53*, 27–51. https://doi.org/10.1111/j.0022-1821.2005.00244.x